

Tipo de documento: Preprint

DUDF: Differentiable Unsigned Distance Fields with Hyperbolic Scaling

Autoría ditelliana: Fainstein, Miguel; Siless, Viviana; Iarussi, Emmanuel

Fecha de publicación: 2024

¿Cómo citar este artículo?

Fainstein, M., Siless, V., Iarussi, E. DUDF: *Differentiable Unsigned Distance Fields with Hyperbolic Scaling*. [Preprint. Universidad Torcuato Di Tella]. Repositorio Digital Universidad Torcuato Di Tella <https://repositorio.utdt.edu/handle/20.500.13098/12380>

El presente documento se encuentra alojado en el Repositorio Digital de la Universidad Torcuato Di Tella bajo una licencia Atribución 4.0 Internacional CC BY 4.0 DEED

Dirección: <https://repositorio.utdt.edu>

DUDF: Differentiable Unsigned Distance Fields with Hyperbolic Scaling

Miguel Fainstein¹

miguelon.f98@gmail.com

Viviana Siless¹

viviana.siless@utdt.edu

Emmanuel Iarussi^{1,2}

emmanuel.iarussi@utdt.edu

¹Universidad Torcuato DiTella, ²CONICET

Abstract

In recent years, there has been a growing interest in training Neural Networks to approximate Unsigned Distance Fields (UDFs) for representing open surfaces in the context of 3D reconstruction. However, UDFs are non-differentiable at the zero level set which leads to significant errors in distances and gradients, generally resulting in fragmented and discontinuous surfaces. In this paper, we propose to learn a hyperbolic scaling of the unsigned distance field, which defines a new Eikonal problem with distinct boundary conditions. This allows our formulation to integrate seamlessly with state-of-the-art continuously differentiable implicit neural representation networks, largely applied in the literature to represent signed distance fields. Our approach not only addresses the challenge of open surface representation but also demonstrates significant improvement in reconstruction quality and training performance. Moreover, the unlocked field’s differentiability allows the accurate computation of essential topological properties such as normal directions and curvatures, pervasive in downstream tasks such as rendering. Through extensive experiments, we validate our approach across various data sets and against competitive baselines. The results demonstrate enhanced accuracy and up to an order of magnitude increase in speed compared to previous methods.

1. Introduction

Surface representation is a fundamental aspect in the field of 3D geometry processing, with explicit methods such as triangle meshes, point clouds, and voxelized representations being traditional choices. Implicit surface representations, on the other hand, have been an integral part of the graphics pipeline for many decades. They encapsulate surfaces as the zero-level set of a function, providing a compact and continuous geometry representation. The novelty in recent years has emerged from parameterizing these implicit functions with Neural Networks (NNs), combining their learning capabilities with the advantages of implicit

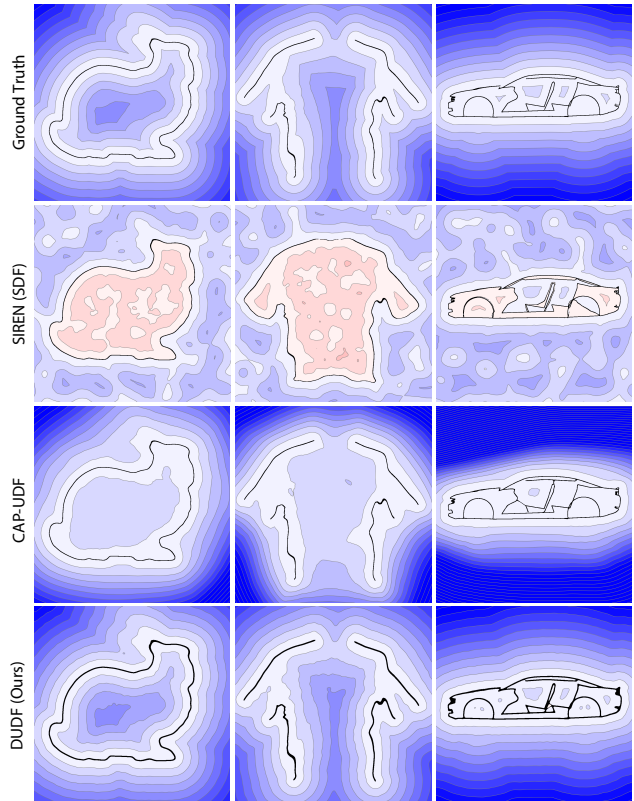


Figure 1. Distance field cross-sections comparative. First row displays ground truth unsigned distance fields. Methods based on signed distances, like SIREN [35], mistakenly fill gaps in open surfaces. While methods based on unsigned distances such as CAP-UDF [45] can represent non-watertight surfaces, the learned distance field do not approximate the true function. DUDF’s differentiable formulation successfully overcomes these challenges.

representations.

Signed Distance Functions (SDFs) have traditionally been the chosen formulation for implicit surface representation [8, 25] due to their well-defined gradients and the ease they offer for computing constructive solid geometry operations and mesh reconstruction. However, they are inherently

limited to closed surfaces (Fig. 1), which poses a significant challenge for representing open surfaces with implicit methods. This limitation arises from the inside/outside sign flip on the zero level set, which is impossible to define for surfaces that do not enclose a volume.

The advent of Unsigned Distance Functions (UDFs) extended representation capabilities to open surfaces. However, this advancement introduced a challenge: the non-differentiable nature of the distance functions at the zero level set. Such non-differentiability leads to inaccuracies in distances and gradients learned by NNs, particularly near the surface where precision is paramount. This paper tackles the challenge of representing open surfaces with neural networks, focusing on a formulation that ensures the surface continuity and smoothness. Our framework (DUDF) combines a hyperbolic scaling of the distance field, together with a new *Eikonal* problem featuring tailored boundary conditions. This formulation facilitates the training of continuously differentiable implicit neural representation networks, while preserving the essential attributes of the unsigned distance fields. Additionally, the differentiable nature of our representations enables precise calculations of crucial topological characteristics, including curvatures and normals, which are pervasive in downstream tasks such as rendering. In contrast, earlier techniques depended on mesh reconstruction methods like Marching Cubes [14, 19, 46] for rendering purposes.

We evaluate our framework with experiments that rigorously assess performance on a variety of challenging data sets and compare to competitive benchmarks in the field. Our method overcomes several limitations of previous approaches, which struggled with the non-differentiability of UDFs. Results demonstrate that our approach not only enhances the reconstruction quality of open surfaces but also accelerates the training process. We believe the dual improvement in precision and efficiency enhance the applicability of neural UDFs representations in geometric processing for real-world scenarios.

2. Related Works

The representation and reconstruction of three-dimensional geometry is a fundamental challenge in the field of computer graphics, vision, and computational geometry. We now briefly discuss the key contributions and methodologies adopted in recent literature to address the main challenges in the field, distinguishing between closed and open surface paradigms.

Closed surfaces. Recent years have seen a surge in the use of Neural Implicit Functions (NIFs) for modeling and reconstructing 3D shapes [39]. These representations are usually implemented with Multi-Layer Perceptrons (MLP) or Convolutional Neural Networks (CNNs), but differ in the learning task [4, 5, 21, 29]. On the one hand, some methods

[4, 5, 21, 22, 29] learn an indicator function or binary occupancies which are later used to reconstruct the 3D surface. On the other hand, alternative methods focus on estimating the SDF at any point in 3D space [9, 11, 15, 23, 24, 27]. The landscape of NIFs has further expanded with the introduction of novel approaches like implicit moving least-squares surfaces [16], a differentiable Poisson solver [28], a complex Gabor wavelet [32], and a level set alignment loss [20]. Notably, central works in the field [13, 35] approached the problem as finding solutions to the *Eikonal* equation alongside expressive boundary conditions, building upon the literature on solving PDEs with neural networks [30, 34]. In particular, SIREN [35] achieved remarkable results utilizing periodic activation functions, allowing to successfully control the function’s differential fields. Despite these advancements, a fundamental limitation of current NIF approaches remains in their inability to represent open surfaces, a characteristic often exhibited by real-world objects like scene walls, clothing, or vehicles with inner structures.

Open surfaces. In order to model general non-watertight surfaces, Chibane et al. introduced the idea of learning UDFs, pioneering the handling of open surfaces with neural networks [6]. Following this, several methods have aimed to improve the performance of open-surface representation using neural networks [36, 37, 41–43, 45]. For instance, GIFS [42] and Neural Vector Fields [41] model the relationships between every two points instead of the relationships between points and surfaces. HSDF separately learns sign and distance fields to handle surfaces with arbitrary topologies [38]. NeuralUDF [18] and NeUDF [17] focused on learning an unsigned distance field by volume rendering for multi-view reconstruction of surfaces with arbitrary topologies. Closer to our work is CAP-UDF [45], which optimized models on raw point clouds by learning to move 3D point queries until reaching the surface with a field consistency constraint. Despite these developments, since UDFs are not differentiable at the zero level set, the gradient field is ill-defined near the isosurface. This leads to difficulties during training, but specially during surface reconstruction, where gradients are often used to determine surface presence between points [14, 46], compute surface normals, and project points onto the isosurface [6] through gradient descent, among others. Recent work by Zhou et al. aimed to solve this issue by introducing constraints at the zero level set in the form of losses leading to a smoother surface [44]. In contrast, our approach formulates an *Eikonal* problem which is solved by learning a scaled distance function that remains differentiable close to the surface. We demonstrate that our method not only enhances reconstruction quality, addressing the prevailing challenges associated with the non-differentiability of the unsigned distance, but also improves *state-of-the-art* in terms of training time performance.

3. Proposed Approach

3.1. Mathematical background

Methods addressing closed surfaces approach the learning of SDFs as finding the solution to a system of Partial Differential Equations (PDEs) governed by the homogeneous *Eikonal* equation with *Dirichlet* and *Neumann* boundary conditions. Formally, given a closed surface \mathcal{S} in $\mathcal{C} \subseteq \mathbb{R}^3$ they seek to find a continuous function f that satisfies $\|\nabla f\| = 1$, with boundary conditions $f|_{\mathcal{S}} = 0$, $\nabla f|_{\mathcal{S}} = \mathbf{n}_{\mathcal{S}}$. Where $\mathbf{n}_{\mathcal{S}}$ denotes the unitary normal field at the surface. Although SDFs are not differentiable at every point and represent only a weak solution to the *Eikonal* equation, recent research has demonstrated considerable success in solving this problem with continuously differentiable implicit neural representation networks [7, 13, 26, 35]. These architectures use periodic activations, facilitating smoother optimization processes and improved control over the solution’s gradient field. This is achievable because locations where the signed distance lacks differentiability are distant from the isosurface, hence the approximation tends to be good in a close neighborhood of the zero level set. In particular, using these networks enable accurate computation of critical topological properties, including mean and Gaussian curvature [26]. This contrasts with networks featuring piecewise linear activations, such as *ReLU*s, which have null second-order derivatives [35].

In the context of representing UDFs, these functions cannot be a solution to the *Eikonal* equation both in the interior and exterior regions of the surface, without losing differentiability at the zero level set. This presents challenges for continuously differentiable implicit neural representation networks in achieving satisfactory outcomes, being that the approximation errors happen at the isosurface where accuracy is paramount. To our knowledge, there has been no successful report of a solution to the *Eikonal* equation accurately approximating unsigned distance functions.

3.2. Problem statement

Our key insight is to overcome these challenges by redefining the unsigned distance field, particularly through the application of an hyperbolic scaling. In light of this understanding, we propose to learn the parameters θ of a neural network f_{θ} with periodic activation functions [35] to approximate the function:

$$t_{\mathcal{S}}(\mathbf{x}) = d_{\mathcal{S}}(\mathbf{x}) \tanh(\alpha d_{\mathcal{S}}(\mathbf{x})), \quad (1)$$

where $d_{\mathcal{S}}$ is the unsigned distance to surface \mathcal{S} ; and α is a constant value. The function $t_{\mathcal{S}}$ is a differentiable approximation of $d_{\mathcal{S}}$, whose zero level set is the surface \mathcal{S} . In Fig. 2 (a), we show the impact of our hyperbolic scaling on the unsigned distance field, mainly characterized by quadratic smoothing in proximity to the isosurface and a

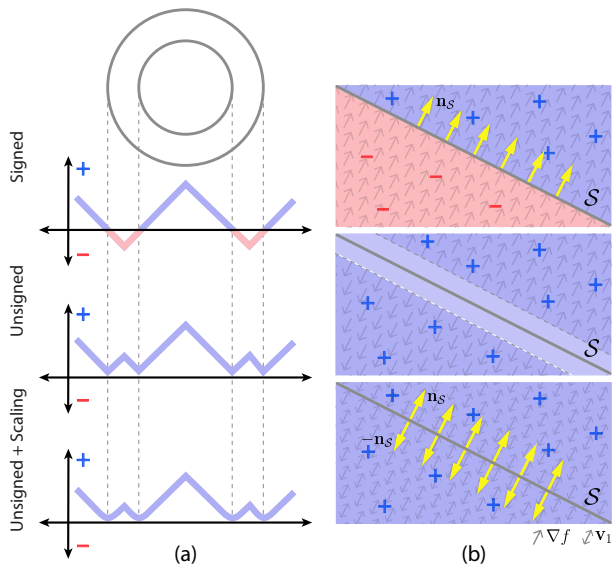


Figure 2. In (a), view of different distance fields for a 2D slice of a torus. Note the effect of hyperbolic scaling near the isosurface (bottom). In (b), sign and gradient for each distance field f . In signed distance, the gradient ∇f at the isosurface is equal to the surface’s normal field $\mathbf{n}_{\mathcal{S}}$ (top). In unsigned distance, the gradient is undefined at the isosurface due to the change in orientation (middle). Our maximum curvature field vectors \mathbf{v}_1 align with the surface’s unoriented normals (bottom).

linear behavior in the distance. The parameter α controls the distance at which the smoothing occurs.

The scaling enables the application of continuously differentiable implicit neural representation networks to solve an *Eikonal* equation, while retaining the UDF’s open surface representation capabilities. For this task, we aim to address an heterogeneous *Eikonal* equation, for which we know $t_{\mathcal{S}}$ (Eq. 1) is a weak solution:

$$\begin{cases} \|\nabla f\| = \phi \\ f|_{\mathcal{S}} = 0 \\ \nabla f|_{\mathcal{S}} = \mathbf{0} \end{cases} \quad (2)$$

where ϕ is defined as the L2 norm of the gradient of Eq. 1, formally:

$$\phi(\mathbf{x}) = \tanh(\alpha d_{\mathcal{S}}(\mathbf{x})) + \alpha d_{\mathcal{S}}(\mathbf{x})(1 - \tanh^2(\alpha d_{\mathcal{S}}(\mathbf{x}))). \quad (3)$$

This formulation assumes that the domain boundary is defined by a closed surface \mathcal{S} , where *Dirichlet* and *Neumann* boundary conditions can be imposed. However, that framework does not directly apply to open surfaces, which do not enclose a well-defined domain and thus cannot support the imposition of boundary constraints. To address this limitation, we reformulate the problem as an initial value problem where the *Dirichlet* condition on \mathcal{S} prescribes the initial values for f , and where the *Neumann* condition can

be omitted given that is already satisfied by the *Eikonal* equation when $\phi(\mathbf{s}) = 0$ for $\mathbf{s} \in \mathcal{S}$.

The problem outlined in Eq. 2 does not incorporate the information of the surface normals. This contrasts with prior research focused on closed surfaces, where leveraging such information has led to improved accuracy and enhanced reconstruction quality by further constraining the set of feasible solutions. Our second key insight is that although the gradient of $t_{\mathcal{S}}$ (Eq. 1) becomes null at the isosurface, the direction in which the gradient norm increases most rapidly remains non-null. This direction is indicated by the maximum curvature of f , defined as the eigenvector associated with the largest eigenvalue of the Hessian matrix. Note that in this context, the maximum curvature does not refer to the surface’s curvature, but rather to the curvature of the hyperbolic scaled unsigned distance field f , where the function’s rate of change varies most rapidly. Formally, $\mathbf{H}_{t_{\mathcal{S}}}$ is the Hessian matrix of $t_{\mathcal{S}}$ which has 3 real eigenvalues $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3|$; be \mathbf{v}_1 a unitary eigenvector associated with λ_1 at \mathbf{s} , it can be shown that $\mathbf{v}_1 = \pm \mathbf{n}_{\mathcal{S}}$ for every $\mathbf{s} \in \mathcal{S}$. Therefore, we further condition the solution to Eq. 2 by adding an extra boundary condition enforcing alignment between the unitary directions associated to the maximum curvature field and the unitary normal field of the surface.

3.3. Implicit function learning

Following the aforementioned definitions, our neural networks are trained to minimize the following loss function:

$$\mathcal{L} = \lambda_e \mathcal{L}_{Eikonal} + \lambda_d \mathcal{L}_{Dirichlet} + \lambda_n \mathcal{L}_{Neumann} + \lambda_g \mathcal{L}_{MCurv}, \quad (4)$$

with λ_i constant weights controlling the relevance of each term. The term which favors a solution to the *Eikonal* PDE is defined as:

$$\mathcal{L}_{Eikonal} = \int_{\mathcal{C}} \left| \|\nabla f_{\theta}(\mathbf{x})\| - \phi(\mathbf{x}) \right| dx. \quad (5)$$

The *Dirichlet* boundary condition loss term controlling function values at the surface is formally defined as:

$$\mathcal{L}_{Dirichlet} = \int_{\mathcal{S}} |f_{\theta}(\mathbf{x})| dx. \quad (6)$$

Similarly to previous works [7, 26], we extend this conditioning to points far from the surface \mathcal{S} . This is achieved by computing an approximation of the function $t_{\mathcal{S}}$ (Eq. 1) based on nearest neighbors. *Neumann’s* boundary condition is expressed in the loss term:

$$\mathcal{L}_{Neumann} = \int_{\mathcal{S}} \|\nabla f_{\theta}(\mathbf{x})\| dx. \quad (7)$$

Finally, we ensure the second-order boundary condition by aligning the directions of maximum curvature with the sur-

face normal through the computation of the following integral:

$$\mathcal{L}_{MCurv} = \int_{\mathcal{S}} 1 - |\mathbf{v}_1(\mathbf{x}) \cdot \mathbf{n}_{\mathcal{S}}(\mathbf{x})| dx, \quad (8)$$

where $\mathbf{v}_1(\cdot)$ is the unitary eigenvector associated with the largest eigenvalue of $\mathbf{H}_{f_{\theta}}(\mathbf{x})$, the Hessian matrix of f_{θ} at \mathbf{x} . In practice, we approximate all integrals discretely using a data set comprised of tuples $\{(\mathbf{x}_i, d_{\mathcal{S}}(\mathbf{x}_i), \mathbf{n}_{\mathcal{S}}(\mathbf{x}_i))\}_i$. At every training iteration, the gradient and Hessian of the neural network are computed through automatic differentiation.

3.4. Isosurface refinement

In pursuit of solutions that exhibit a consistent, near-zero value at the isosurface —essential for smooth reconstructions— we further fine-tune our networks in a second optimization step. Formally, we minimize the following loss function:

$$\mathcal{L}_{Refinement} = \lambda_{\mu} |\mu(f_{\theta|_{\mathcal{S}}})| + \lambda_{\sigma} \sigma(f_{\theta|_{\mathcal{S}}}), \quad (9)$$

where $\mu(\cdot)$ is the mean value and $\sigma(\cdot)$ the standard deviation.

By minimizing the mean and variance of the learned function values at the isosurface, we ensure they are as close to zero as possible. This refinement process effectively reduces the oscillations and deviations that can occur at the critical boundary, thereby enhancing the overall quality of the reconstructed surfaces.

3.5. Normals and curvature computation

A significant advantage of our formulation over prior efforts [6, 46] is the fact that gradient fields and higher-order derivatives are well-defined in the vicinity of the isosurface, allowing the direct computation of topological properties during inference and reconstruction. This means that we are able to render unsigned distance fields using standard algorithms such as sphere tracing. However, directly using the gradients of f_{θ} at the isosurface as normals for rendering purposes can be unreliable since their norm is close to 0. Hence, for a given surface value threshold ϵ , when a point \mathbf{s} is identified such that $f_{\theta}(\mathbf{s}) < \epsilon$, we determine the normal direction to the surface by calculating the unit eigenvectors associated with the maximum eigenvalue of $\mathbf{H}_{f_{\theta}}(\mathbf{s})$. Note that there are two unitary eigenvectors per surface point, \mathbf{v}_1 and $-\mathbf{v}_1$; however this is usually the desired behavior since open surfaces may not be orientable, such as the *möbius strip*. To address this ambiguity in the sign we consider the position of the camera when rendering. Additionally, other relevant topological properties such as the mean and Gaussian curvature are also available by means of the maximum unit eigenvector field. See Sec. 4.6 for more details.

4. Results and evaluation

4.1. Experimental setup

We conducted a series of experiments to assess the performance of our method. To target a broad amount of surfaces, we experimented on three well-known data sets: ShapeNet cars [3], Multi-Garment [2], and DeepFashion [48]. For each data set we trained individual networks on 30 randomly selected oriented point clouds. For the Deep Fashion and Multi-Garment dataset, we sampled 100,000 surface points, while for ShapeNet cars we doubled the sample size to account for the higher degree of complexity. On each training iteration, the loss was computed on a subset of 30,000 points, equally distributed into three distinct groups: surface points (1), far domain points (2), and near domain points (3). Surface points (1) were randomly selected from the ground truth point cloud. Far points (2) were uniformly generated within the function’s domain: a cube with side length 2 and centered at the origin. All shapes were uniformly scaled to fit within this volume before sampling. The distance to the point clouds was approximated using a tree-based search algorithm [47]. Finally, since we found that a biased near-the-surface sampling improved accuracy, we used near domain points (3) during training. We constructed this set by randomly displacing the sampled surface points in their normal direction. Displacements were sampled from a normal distribution centered at zero with a standard deviation of 0.01. The corresponding unsigned distance was approximated with the distance to the undisplaced surface point. This was experimentally found to be sufficiently accurate and fast.

4.2. Network architecture

Our network takes spatial coordinates x, y, z as input, and outputs the predicted t_S value from Eq. 1. The architecture consists of an 8-layer 256-units MLP with *sine* activations. The same number of layers and parameter count were used for baseline methods. Loss weights were experimentally set to $\lambda_e, \lambda_d, \lambda_n = 1e^4$, $\lambda_g = 1e^3$ and $\lambda_\mu, \lambda_\sigma = 1e^5$. Models were implemented in Python using Pytorch library, and distance metrics were computed using Pytorch3D [31]. We trained each network for 3,000 iterations, the first third of them utilized a learning rate of $1e^{-4}$, in the following third we lowered it to $1e^{-5}$, and in the final iterations (only the refinement loss active) we set the learning rate to $1e^{-7}$ with a cosine decay. These final small learning rates allowed the refinement of the isosurface without losing the gained accuracy in the rest of the field. Regarding the value of α (Eq. 1), we experimentally found that choosing a value of 100 kept a good balance between training accuracy and reconstruction quality. Experiments were run on Ubuntu, an Intel(R) Xeon(R) Silver 4310 CPU, 256Gb of RAM, and 2 Nvidia A100 graphics cards of 80Gb VRAM.

4.3. Mesh reconstruction

For a thorough comparison of our method against existing techniques, we undertake mesh reconstruction using two gradient-based Marching Cubes algorithms, referred here as MC1 [46] and MC2 [14]. These methods perform linear interpolation between grid values for accurate placement of triangle vertices. Given this requirement and considering that the function t_S , as described in Equation 1, exhibits quadratic behavior near the isosurface, we retrieve the true unsigned distance by first dividing by α and then applying the square root. For a comprehensive analysis we include SIREN [35], which does not approximate an unsigned distance, as illustrated in Figure 1. In this case, we used traditional Marching Cubes [19] for the reconstruction.

4.4. Surface reconstruction

4.4.1 Closed shapes

We benchmarked DUDF against state-of-the-art methods for representing closed surfaces, where signed distances are well-defined. Given that our primary focus is not on closed surface representation, we limited this comparison to the ShapeNet car dataset [3], modifying the meshes for closure and omitting internal structures [40]. The comparative results are detailed in Table 1. While DeepSDF [27] excels in speed, it falls short in reconstruction quality. Being primarily designed for shape generation, DeepSDF is outperformed by methods tailored for accurate single shape representation, as evidenced by the Chamfer distance metrics between ground truth and reconstructed point clouds.

SIREN [35] offers improved shape representation with enough detail and smoothness. However, since its zero level set satisfies the implicit function theorem, it cannot properly represent shapes with right angles and thin structures. In contrast, CAP-UDF [46] can better approximate these sharp features because it does not rely on continuously differentiable implicit neural representation networks, thereby achieving higher accuracy. However, the training time is high and the reliance on gradient based Marching Cubes methods for surface reconstruction tends to create surfaces

Method	time(s) ↓	L1CD ↓	L2CD ↓
DeepSDF [27]	94	19.40	0.206
SIREN [35]	379	15.40	0.171
CAP-UDF [46]	1080	9.48	0.030
Ours (MC1 [46])		9.48	0.028
Ours (MC2 [14])	319	9.49	0.028

Table 1. Training time, L1, and L2 mean Chamfer distances ($\times 10^3$) for the closed ShapeNet cars data set. DeepSDF and SIREN were trained on signed distances, while CAP-UDF and our method were trained on unsigned distances.

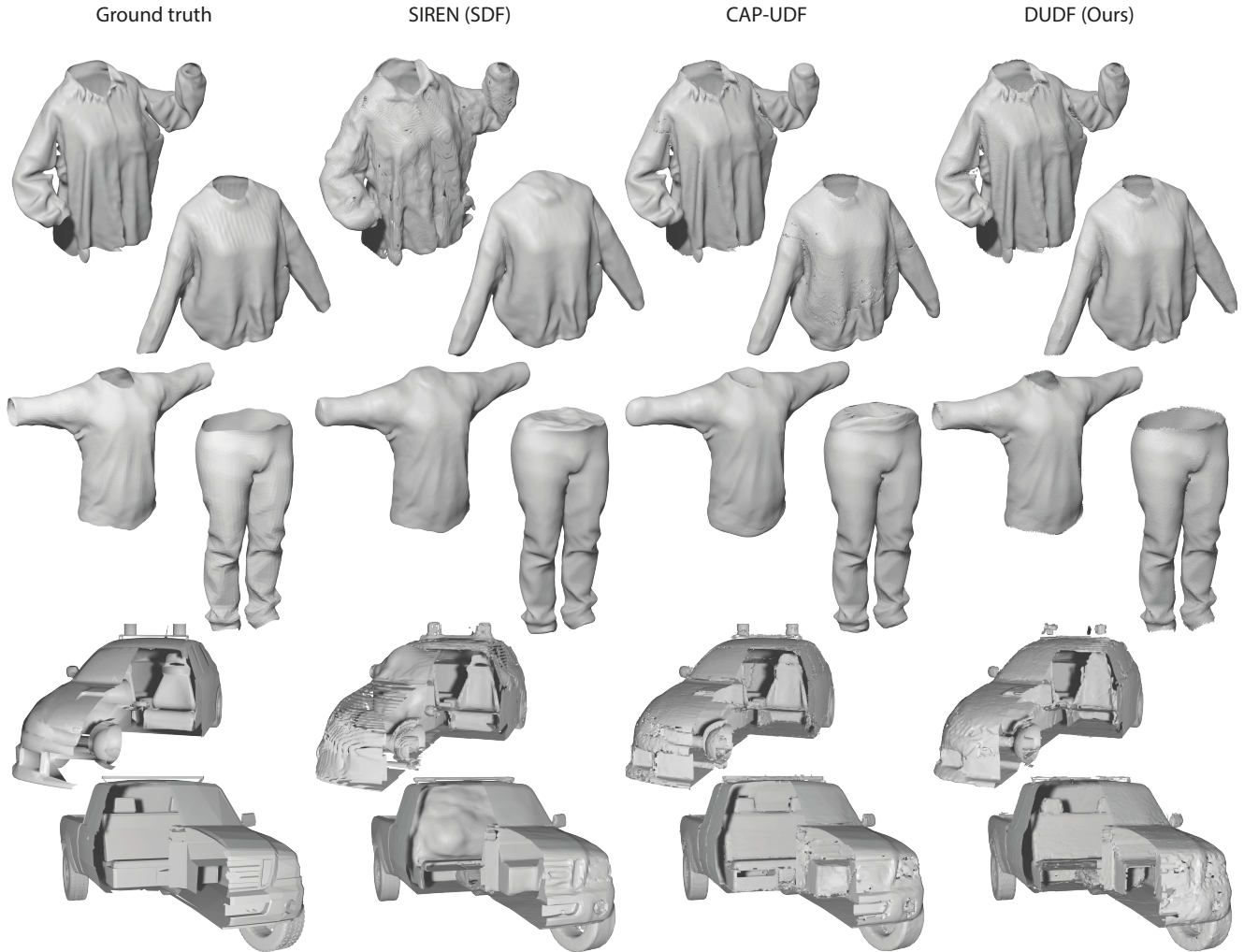


Figure 3. Comparisons on DeepFashion [48] (top row), Multi-Garment [2] (middle row), and ShapeNet cars [3] (bottom row) data sets. DUDF preserves fine details and accurately represents complex geometries without closing holes, outperforming SIREN (SDF), which tends to smooth and round models, and CAP-UDF, which captures sharp features but often closes open surfaces. Reconstructions for CAP-UDF and DUDF performed with MCI [46].

with small holes and discontinuities which hinders the visual quality in some cases. Our approach captures a differentiable function’s zero level set like SIREN, but uniquely bypasses the implicit function theorem by having null gradients at the isosurface. This allows our method to effectively represent sharp features and complex geometries, leading to improved precision in surface reconstruction, particularly in scenarios where surfaces are not inherently smooth.

4.4.2 Open surfaces

When evaluating open-surface representation, we primarily compare our method against CAP-UDF [46]. This approach was shown to be more effective than earlier techniques (i.e. NDF [6]), that required a complex process for creating a

dense point cloud and reconstructing a triangulated surface using the Ball Pivoting algorithm [1], which lead to poor quality meshes with great amounts of holes and irregularities. Additionally, a recent method by Zhou et al. [44] has introduced constraints on the zero level set, which the authors claim it leads to better reconstructions. However, the absence of available implementation code in their official repository precluded a fair comparison with our method. Therefore, we concentrate our analysis on the most recent and available methodologies.

Qualitative comparisons can be observed in Fig. 3 and a full quantitative analysis is presented in Table 2. On the one hand, our method demonstrates a significant improvement in efficiency, consuming up to an order of magnitude less computational time than CAP-UDF, while also show-

Method	DeepFashion [48]			Multi-Garment [2]			ShapeNet cars [3]		
	time(s) ↓	L1CD ↓	L2CD ↓	time(s) ↓	L1CD ↓	L2CD ↓	time(s) ↓	L1CD ↓	L2CD ↓
SIREN [35]	376	27.3	1.980	374	40.5	8.810	751	16.9	0.170
CAP-UDF [46]	1390	18.1	1.110	1440	18.5	1.190	1040	10.9	0.071
Ours (MC1 [46])	326	9.01	0.025	318	8.70	0.024	317	12.3	0.057
Ours (MC2 [14])		9.14	0.027		8.82	0.026		13.7	0.081

Table 2. Training time, L1, and L2 mean Chamfer distances ($\times 10^3$) for the evaluated open surface data sets. SIREN was trained as described in the original paper, while CAP-UDF and our method were trained on unsigned distances.

ing enhanced performance across all three data sets. This advantage is particularly evident in the DeepFashion and Multi-Garment data sets, where CAP-UDF tends to inaccurately close openings (like those at the ends of sleeves in clothing). On the other hand, since SIREN was devised for closed surfaces, it creates a negative distance valued shell around the isosurface. This causes to either close every hole or a thickened surface enclosing the ground-truth level set.

In addition to these experiments, we compare our method with CAP-UDF in the context of sphere tracing rendering. This comparison is presented in Fig. 4, where we showcase two rendering examples. Since learned unsigned distance fields in CAP-UDF do not grow linearly away from the surface, the marching steps in sphere tracing often fail to accurately intersect the surface, leading to undesirable visual artifacts. Even when hitting the surface, gradients might be undefined, leading to noisy shaded images. In contrast, our proposed distance function does not exhibit these problems, offering a more robust framework for direct rendering. Furthermore, our ability to compute normals using the maximum curvature field is useful for shading purposes, enhancing the rendering quality without undergoing an intermediate 3D mesh reconstruction step.

4.5. Ablation study

We now dissect the impact of individual loss components on the network’s accuracy through an ablation study. The methodology involves selectively deactivating terms in our loss function (Eq. 4) to isolate their contributions. We adopted the experimental framework from the prior section,

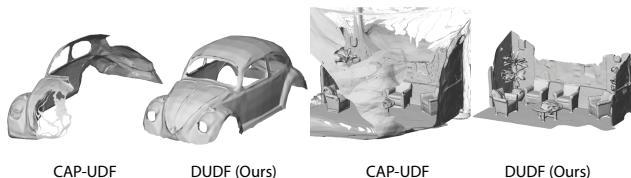


Figure 4. Rendering examples using sphere tracing. CAP-UDF struggles with non-linear growth of their unsigned distance fields, causing the sphere tracing marching step to miss the surface. Conversely, our method demonstrates precision in direct rendering scenarios.

training on 30 meshes from the DeepFashion dataset [48]. To elucidate the role of boundary conditions in enhancing reconstruction quality, we incorporated a normal consistency metric into our analysis. This metric is computed as the mean value of the absolute cosine similarities between each point to its closest neighbor in the other point cloud. We report metrics for reconstructions performed using MC2 [14], yielding results analogous to those obtained with MC1 [46]. Additionally, we contrasted the effects of approximating the distance function d_S with t_S (Eq. 1), utilizing sine and *ReLU* activation functions (each with distinct *Eikonal* problems).

The findings presented in Table 3 underscore the significance of both the *Eikonal* equation (Eq. 5) and the *MCurv* boundary condition (Eq. 8) in achieving superior normal consistency, which notably influences the rendering quality. Moreover, the refinement step during training is an important element for normal consistency improvement, offering substantial increased performance for a small additional computational cost.

4.6. Computing mean and Gaussian curvature

A significant advantage of our method over previous approaches is the possibility to compute curvatures. Previous work learned non-differentiable functions, which preclude the direct computation of geometrical properties such as mean and Gaussian curvature. Our differentiable UDFs facilitates the extraction of these curvature values. Leverag-

Method	time(s) ↓	L1CD ↓	L2CD ↓	NC ↓
Baseline	326	9.14	0.027	0.020
$\lambda_e = 0$	312	9.43	0.028	0.033
$\lambda_g = 0$	150	9.16	0.027	0.021
$\lambda_e, \lambda_g = 0$	109	9.44	0.027	0.035
$\lambda_\mu, \lambda_\sigma = 0$	302	9.24	0.027	0.031
d_S (sine)	150	31.2	0.830	0.057
d_S (<i>ReLU</i>)	145	47.5	2.500	0.149

Table 3. Quantitative impact of each loss component on the reconstruction accuracy of our network. We report training time, L1, and L2 mean Chamfer distances $\times 10^3$, and Normal Consistency (NC).

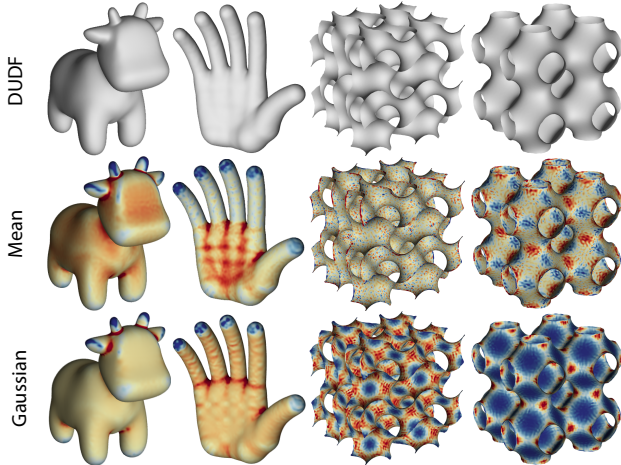


Figure 5. Mean and Gaussian curvatures computed with our method for closed surfaces (left) and open surfaces (right).

ing the divergence ($\nabla \cdot$) of the normal field (\mathbf{n}_S), the mean curvature H at any point on the surface \mathbf{s} is then computed [10, 12, 26] as:

$$H(\mathbf{s}) = \frac{1}{2} \nabla \cdot \mathbf{n}_S(\mathbf{s}). \quad (10)$$

Additionally, the Gaussian curvature K can be computed using the determinant of a matrix composed of the Jacobian matrix $J_{\mathbf{n}_S}(\mathbf{s})$ of the unit normal field (\mathbf{n}_S) along with the normal field itself. Formally, we compute:

$$K(\mathbf{s}) = -\det \left[\begin{array}{c|c} J_{\mathbf{n}_S}(\mathbf{s}) & \mathbf{n}_S(\mathbf{s}) \\ \hline \mathbf{n}_S(\mathbf{s})^t & 0 \end{array} \right]. \quad (11)$$

In Fig. 5 we show mean and Gaussian curvatures for open and closed surfaces computed with our method.

5. Limitations and future work

Our method encounters certain limitations. Primarily, by approximating the function t_S (Eq. 1), the *Eikonal* equation requires supervision with ground truth unsigned distances, in contrast to signed distance approaches where it is constant everywhere. In addition, the quadratic behavior of the function near the isosurface produces a wider near-zero strip which can lead to slightly inflated representations. This effect can be lessened by computing square roots, at the expense of an increased computational cost. Secondly, we observed that surfaces extracted using gradient-based Marching Cubes methods can sometimes be less smooth than directly rendering the function using sphere tracing. This is likely due to the current reconstruction algorithm’s sensitivity to noise, an issue that might be addressed by employing a tangency-aware surface reconstruction strategy [33]. Additionally, computing the maximum curvature direction

field is computationally expensive, requiring three network passes and the diagonalization of the Hessian matrix. However, future improvements in the network’s design could reduce the number of parameters, thus lessening the computational burden associated with computing topological features.

6. Conclusion

In this work we introduced Differentiable Unsigned Distance Fields (DUDF) with Hyperbolic Scaling, a novel approach that addresses inherent limitations of traditional UDFs in representing open surfaces. By applying a hyperbolic transformation to the distance field, we define a new variant of the *Eikonal* problem, tailored with unique boundary conditions. This adaptation enables our model to capitalize on the robust framework of continuously differentiable neural networks, thus enhancing reconstruction quality and training efficiency. The conducted experiments provide evidence that our approach can lead to improvements in reconstruction quality and computational performance when compared to several state-of-the-art methods. Moreover, the ability to accurately calculate topological features such as normals and curvatures is a notable benefit of our model, offering additional utility in geometric processing tasks and rendering. While our results are promising, showing better accuracy and efficiency in most cases, we acknowledge the need for further research and development. The potential applications of DUDF in 3D geometry processing are broad, and we are optimistic about its utility in practical scenarios. Future work will focus on exploring these applications and enhancing the model’s performance.

7. Acknowledgments

This project was supported by grants from Salesforce, USA (Einstein AI 2020), National Scientific and Technical Research Council (CONICET), Argentina (PIP 2021-2023 GI -11220200102981CO), and Universidad Torcuato Di Tella, Argentina. We also thank Facundo Fainstein and Tomas Delgado for valuable discussions on the *Eikonal* problem formulation.

References

- [1] Fausto Bernardini, J. Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, 5:349 – 359, 11 1999. 6
- [2] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 5, 6, 7
- [3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese,

- Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5, 6, 7
- [4] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2
- [5] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6970–6981, 2020. 2
- [6] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020. 2, 4, 6
- [7] Mattéo Clémot and Julie Digne. Neural skeleton: Implicit neural representation away from the surface. *Computers & Graphics*, pages 368–378, 2023. 3, 4
- [8] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 1
- [9] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. On the effectiveness of weight-encoded neural implicit 3d shapes. *arXiv preprint arXiv:2009.09808*, 2020. 2
- [10] Manfredo P Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016. 8
- [11] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. Curriculum deepsf. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 51–67. Springer, 2020. 2
- [12] Ron Goldman. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design*, 22(7):632–658, 2005. 8
- [13] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 2
- [14] Benoit Guillard, Federico Stella, and Pascal Fua. Meshudf: Fast and differentiable meshing of unsigned distance field networks, 2022. 2, 5, 7
- [15] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 2
- [16] Shi-Lin Liu, Hao-Xiang Guo, Hao Pan, Peng-Shuai Wang, Xin Tong, and Yang Liu. Deep implicit moving least-squares functions for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1788–1797, 2021. 2
- [17] Yu-Tao Liu, Li Wang, Jie Yang, Weikai Chen, Xiaoxu Meng, Bo Yang, and Lin Gao. Neudf: Leaning neural unsigned distance fields with volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 237–247, 2023. 2
- [18] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Yuan Liu, Peng Wang, Christian Theobalt, Taku Komura, and Wenping Wang. Neuraludf: Learning unsigned distance fields for multi-view reconstruction of surfaces with arbitrary topologies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20834–20843, 2023. 2
- [19] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, volume 21, pages 347–353. 1998. 2, 5
- [20] Baorui Ma, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Towards better gradient consistency for neural signed distance functions via level set alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17724–17734, 2023. 2
- [21] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2
- [22] Zhenxing Mi, Yiming Luo, and Wenbing Tao. Ssrnet: Scalable 3d surface reconstruction network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 970–979, 2020. 2
- [23] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv preprint arXiv:1901.06802*, 2019. 2
- [24] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019. 2
- [25] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 1
- [26] Tiago Novello, Guilherme Schardong, Luiz Schirmer, Vinicius da Silva, Helio Lopes, and Luiz Velho. Exploring differential geometry in neural implicits, 2022. 3, 4, 8
- [27] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2, 5
- [28] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *Advances in Neural Information Processing Systems*, 34:13032–13044, 2021. 2
- [29] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2

- [30] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations, 2017. [2](#)
- [31] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. [5](#)
- [32] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023. [2](#)
- [33] Silvia Sellán, Christopher Batty, and Oded Stein. Reach for the spheres: Tangency-aware surface reconstruction of sdf. *arXiv preprint arXiv:2308.09813*, 2023. [8](#)
- [34] Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, pages 1339–1364, 2018. [2](#)
- [35] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. [1](#), [2](#), [3](#), [5](#), [7](#)
- [36] Rahul Venkatesh, Tejan Karmali, Sarthak Sharma, Aurobrata Ghosh, R Venkatesh Babu, László A Jeni, and Maneesh Singh. Deep implicit surface point prediction networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12653–12662, 2021. [2](#)
- [37] Bing Wang, Zhengdi Yu, Bo Yang, Jie Qin, Toby Breckon, Ling Shao, Niki Trigoni, and Andrew Markham. Rangeudf: Semantic surface reconstruction from 3d point clouds. *arXiv preprint arXiv:2204.09138*, 2022. [2](#)
- [38] Li Wang, Weikai Chen, Xiaoxu Meng, Bo Yang, Jintao Li, Lin Gao, et al. Hsdf: Hybrid sign and distance field for modeling surfaces with arbitrary topologies. *Advances in Neural Information Processing Systems*, 35:32172–32185, 2022. [2](#)
- [39] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022. [2](#)
- [40] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *CoRR*, abs/1905.10711, 2019. [5](#)
- [41] Xianghui Yang, Guosheng Lin, Zhenghao Chen, and Luping Zhou. Neural vector fields: Implicit representation by explicit learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16727–16738, 2023. [2](#)
- [42] Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. Gifs: Neural implicit function for general shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12829–12839, 2022. [2](#)
- [43] Fang Zhao, Wenhao Wang, Shengcai Liao, and Ling Shao. Learning anchored unsigned distance functions with gradient direction alignment for single-view garment reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12674–12683, 2021. [2](#)
- [44] Junsheng Zhou, Baorui Ma, Shujuan Li, Yu-Shen Liu, and Zhizhong Han. Learning a more continuous zero level set in unsigned distance fields through level set projection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3181–3192, 2023. [2](#), [6](#)
- [45] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-aware unsigned distance functions progressively from raw point clouds. *Advances in Neural Information Processing Systems*, 35:16481–16494, 2022. [1](#), [2](#)
- [46] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-aware unsigned distance functions progressively from raw point clouds. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. [2](#), [4](#), [5](#), [6](#), [7](#)
- [47] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. [5](#)
- [48] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. *CoRR*, 2020. [5](#), [6](#), [7](#)

Supplementary material

DUDF: Differentiable Unsigned Distance Fields with Hyperbolic Scaling

Miguel Fainstein¹
firstauthor@il.org

Viviana Siless¹
viviana.siless@utdt.edu

Emmanuel Iarussi^{1,2}
emmanuel.iarussi@utdt.edu

¹Universidad Torcuato DiTella, ²CONICET

1. Differentiability analysis

Given a smooth surface \mathcal{S} , we want to study the differentiability of function:

$$t_{\mathcal{S}}(\mathbf{x}) = d_{\mathcal{S}}(\mathbf{x}) \tanh(\alpha d_{\mathcal{S}}(\mathbf{x})). \quad (1)$$

Notice that function $d_{\mathcal{S}}$ is differentiable almost everywhere outside the surface, hence $t_{\mathcal{S}}$ is trivially differentiable at such points and its gradient is given by:

$$\nabla t_{\mathcal{S}}(\mathbf{x}) = \nabla d_{\mathcal{S}}(\mathbf{x}) \phi(\mathbf{x}), \quad (2)$$

with $\phi(\mathbf{x}) = \tanh(\alpha d_{\mathcal{S}}(\mathbf{x})) + \alpha d_{\mathcal{S}}(\mathbf{x})(1 - \tanh^2(\alpha d_{\mathcal{S}}(\mathbf{x})))$. However, at the isosurface $d_{\mathcal{S}}$ is not differentiable. To show $t_{\mathcal{S}}$ differentiable at $\mathbf{s} \in \mathcal{S}$, we consider the partial derivatives expressed by the limit:

$$\lim_{h \rightarrow 0} \frac{t_{\mathcal{S}}(\mathbf{s} + \mathbf{e}_i h) - t_{\mathcal{S}}(\mathbf{s})}{h} = \lim_{h \rightarrow 0} \frac{t_{\mathcal{S}}(\mathbf{s} + \mathbf{e}_i h)}{h}, \quad (3)$$

where \mathbf{e}_i is the i -th canonical vector. It follows that $d_{\mathcal{S}}(\mathbf{s} + \mathbf{e}_i h) \leq |h|$. Considering that function \tanh is monotonically increasing, then:

$$\left| \frac{t_{\mathcal{S}}(\mathbf{s} + \mathbf{e}_i h)}{h} \right| \leq \left| \frac{|h| \tanh(\alpha |h|)}{h} \right| = |\tanh(\alpha |h|)| \xrightarrow{h \rightarrow 0} 0. \quad (4)$$

Hence, partial derivatives at the isosurface are null. To finish the proof, we show that the hyperplane defined by the partial derivatives at \mathbf{s} correctly approximates the function $t_{\mathcal{S}}$, which is satisfied if the following limit approaches 0:

$$\lim_{\mathbf{x} \rightarrow \mathbf{s}} \frac{|t_{\mathcal{S}}(\mathbf{x}) - t_{\mathcal{S}}(\mathbf{s}) - \nabla t_{\mathcal{S}}(\mathbf{s}) \cdot (\mathbf{x} - \mathbf{s})|}{\|\mathbf{x} - \mathbf{s}\|} = \lim_{\mathbf{x} \rightarrow \mathbf{s}} \frac{|t_{\mathcal{S}}(\mathbf{x})|}{\|\mathbf{x} - \mathbf{s}\|}. \quad (5)$$

Similar to the partial derivative calculation, we bound this limit considering that the distance of every point to the surface must always be smaller or equal to the euclidean distance between a point and some surface point, that is $d_{\mathcal{S}}(\mathbf{x}) \leq \|\mathbf{x} - \mathbf{s}\|$ for every $\mathbf{s} \in \mathcal{S}$. Using this fact in Eq. 5:

$$\frac{|t_{\mathcal{S}}(\mathbf{x})|}{\|\mathbf{x} - \mathbf{s}\|} \leq \frac{\|\mathbf{x} - \mathbf{s}\| |\tanh(\alpha d_{\mathcal{S}}(\mathbf{x}))|}{\|\mathbf{x} - \mathbf{s}\|} = |\tanh(\alpha d_{\mathcal{S}}(\mathbf{x}))| \xrightarrow{\mathbf{x} \rightarrow \mathbf{s}} 0, \quad (6)$$

concluding that function $t_{\mathcal{S}}$ is differentiable at the isosurface.

2. Gradient norm

Based on the results from Sec. 1, at points outside the isosurface where $d_{\mathcal{S}}$ is differentiable, the unsigned distance gradient norm is unitary: $\|\nabla d_{\mathcal{S}}(\mathbf{x})\| = 1$. This is a standard property of distance functions. Given that ϕ is always positive away from the isosurface, the norm of the gradient can be expressed as:

$$\|\nabla t_{\mathcal{S}}(\mathbf{x})\| = \|\nabla d_{\mathcal{S}}(\mathbf{x})\| \phi(\mathbf{x}) = \phi(\mathbf{x}). \quad (7)$$

As previously discussed, at the isosurface the gradient $\nabla t_{\mathcal{S}}(\mathbf{s}) = \mathbf{0}$, then its norm is null. These results explain the choice for the *Eikonal* problem and *Neumann* boundary condition expressed in the paper.

3. Qualitative results

In this section we show additional qualitative comparisons for three different experiments showcased in the paper. In Fig. 1 we demonstrate the effect of the maximum curvature field alignment loss on two reconstructed examples. In Fig. 2 we compare closed surface representations trained on the ShapeNet Cars data set [3] pre-processed by [8] to be closed. In Fig. 3 we show how our field's normal directions and principal curvatures are useful in the context of tracing rendering algorithms and light reflection models. Finally, in Fig. 4 we further compare the different reconstruction algorithms used throughout this work. All reconstructions were performed with the parameters reported in the paper (architecture, parameter count, loss weights, and training scheme).

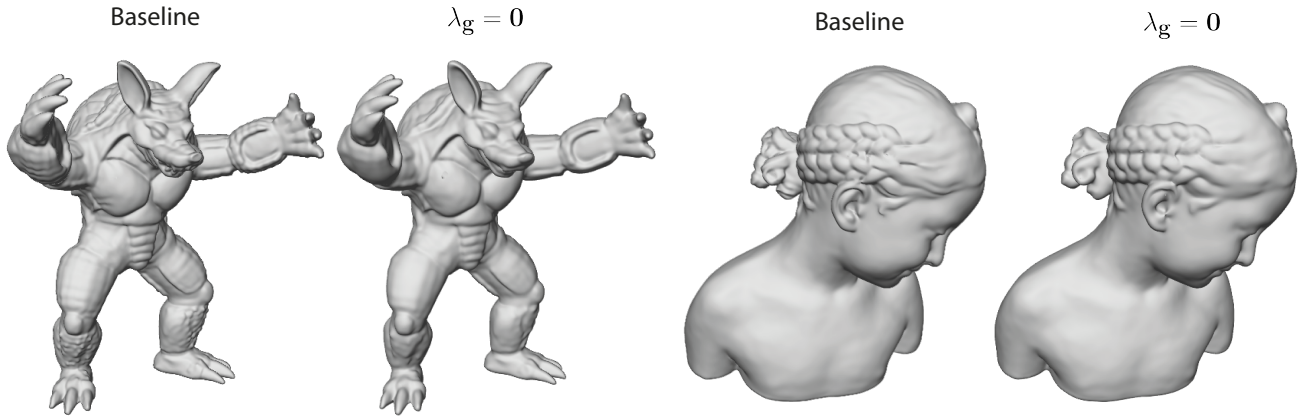


Figure 1. Ablation study. The first column shows the reconstruction of the network trained with the full loss. The second column shows the reconstruction without the maximum curvature field alignment term, resulting in less detail at high curvature regions such as Armadillo’s mouth and Bimba’s hair bun.

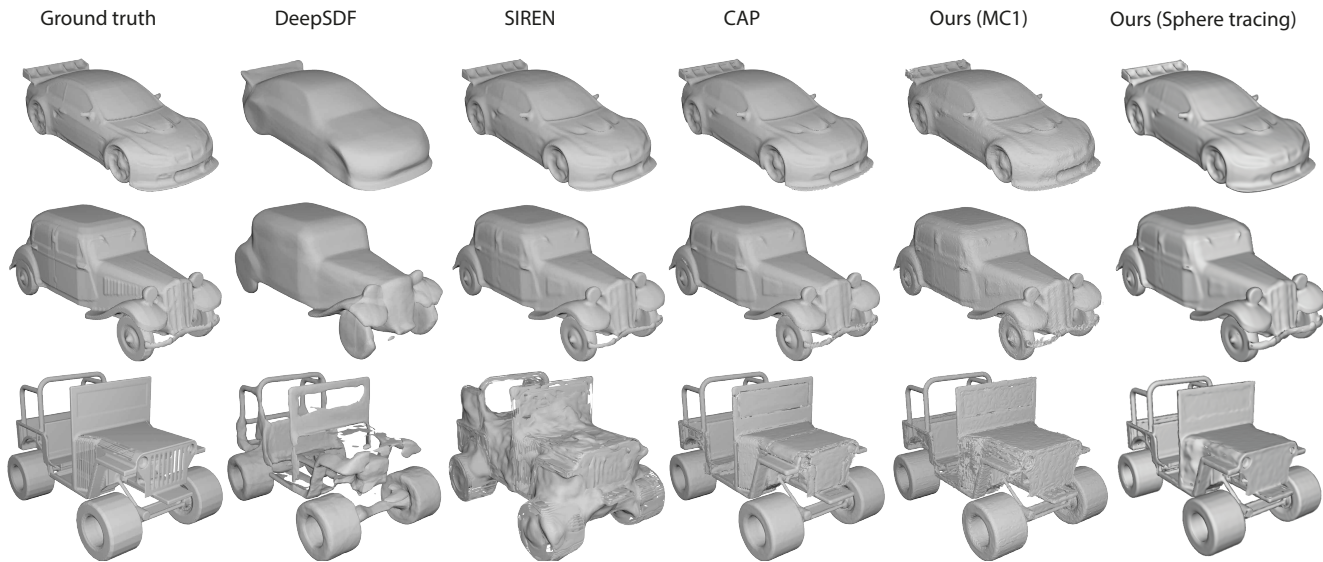


Figure 2. Reconstructions on closed [8] ShapeNet cars [3] for DeepSDF [5], SIREN [6], CAP-UDF [9] and DUDF. We additionally show sphere tracing renderings of our learned fields without the marching cubes reconstruction step.

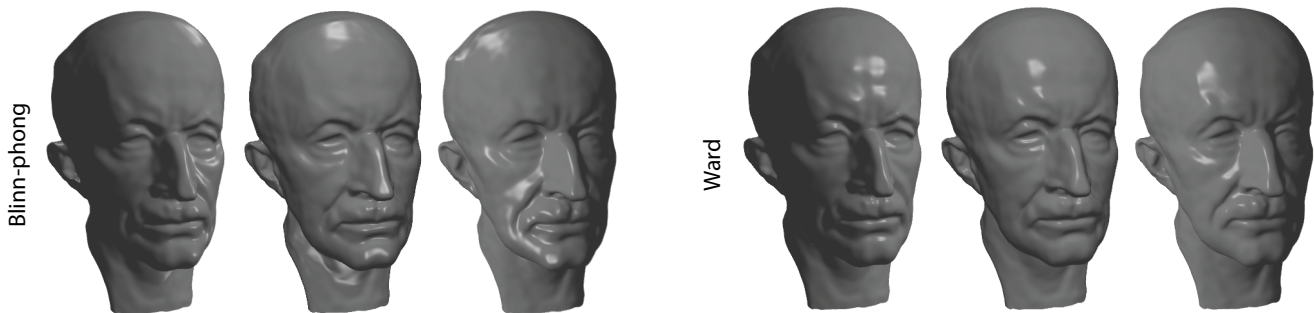
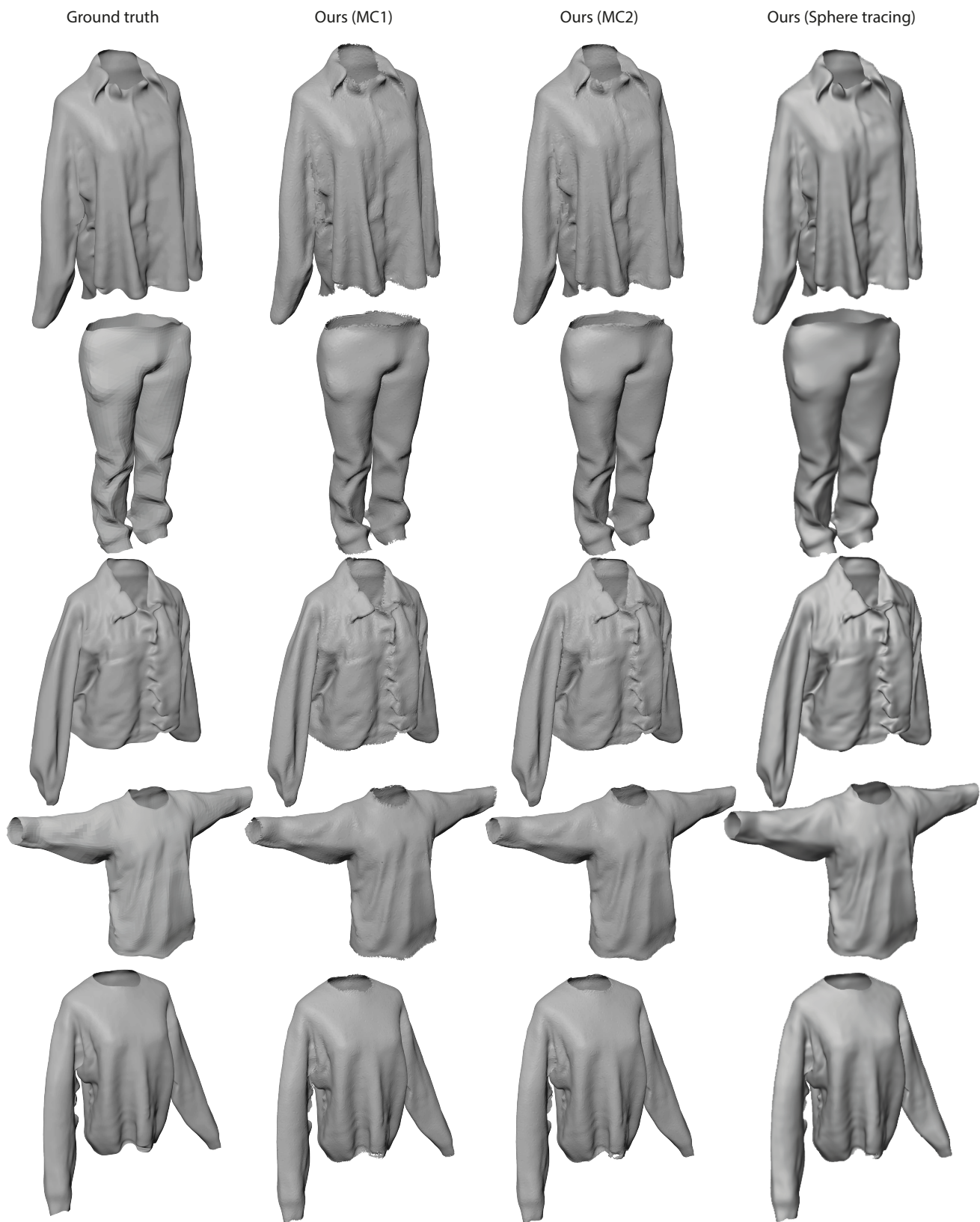


Figure 3. Sphere tracing renderings of an open surface (Max Planck model is open at the bottom) under three different light sources. We used surface normals for the Blinn-phong reflectance model [2]. For the Ward reflectance model [7] we used both, normals and principal curvature directions computed from DUDF’s differentiable fields.



0

Figure 4. Reconstructions on DeepFashion [10] and Multi-garment [1] data sets for the three different reconstruction methods explored in our work: MC1 [9], MC2 [4], and sphere tracing.

References

- [1] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.
- [2] James F. Blinn. Models of light reflection for computer synthesized pictures. page 192–198, 1977.
- [3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
- [4] Benoit Guillard, Federico Stella, and Pascal Fua. Meshudf: Fast and differentiable meshing of unsigned distance field networks, 2022.
- [5] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.
- [6] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.
- [7] Gregory J. Ward. Measuring and modeling anisotropic reflection. 26(2):265–272, 1992.
- [8] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *CoRR*, abs/1905.10711, 2019.
- [9] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-aware unsigned distance functions progressively from raw point clouds. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [10] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. *CoRR*, 2020.